



JACKSON LAWRENCE
00000070612

LOUIS GABRIEL HERNANDES
00000070250

TABU SEARCH

ALGORITHM

OSCAR JIRO HARLISON
00000072786

IGNATIUS STEVEN
00000070642



A

Tabu Search Algorithm

```
Fungsi TabuSearch(masalah, max_iterasi):
```

```
    solusi_saat_ini = solusi_awal  
    nilai_terbaik = nilai(solusi_saat_ini)  
    tabu_list = list kosong
```

```
Untuk i dari 1 sampai max_iterasi:
```

```
    tetangga = generate_tetangga(solusi_saat_ini)  
    tetangga_terbaik = tetangga pertama  
    nilai_terbaik_tetangga = nilai(tetangga_terbaik)
```

```
Untuk setiap tetangga t dalam tetangga:
```

```
    Jika nilai(t) < nilai_terbaik_tetangga dan t tidak dalam tabu_list:  
        tetangga_terbaik = t  
        nilai_terbaik_tetangga = nilai(t)
```

```
solusi_saat_ini = tetangga_terbaik
```

```
Jika nilai_terbaik_tetangga < nilai_terbaik:  
    nilai_terbaik = nilai_terbaik_tetangga
```

```
tambahkan solusi_saat_ini ke tabu_list  
hapus elemen tertua dari tabu_list
```

```
Kembalikan solusi_terbaik
```

B

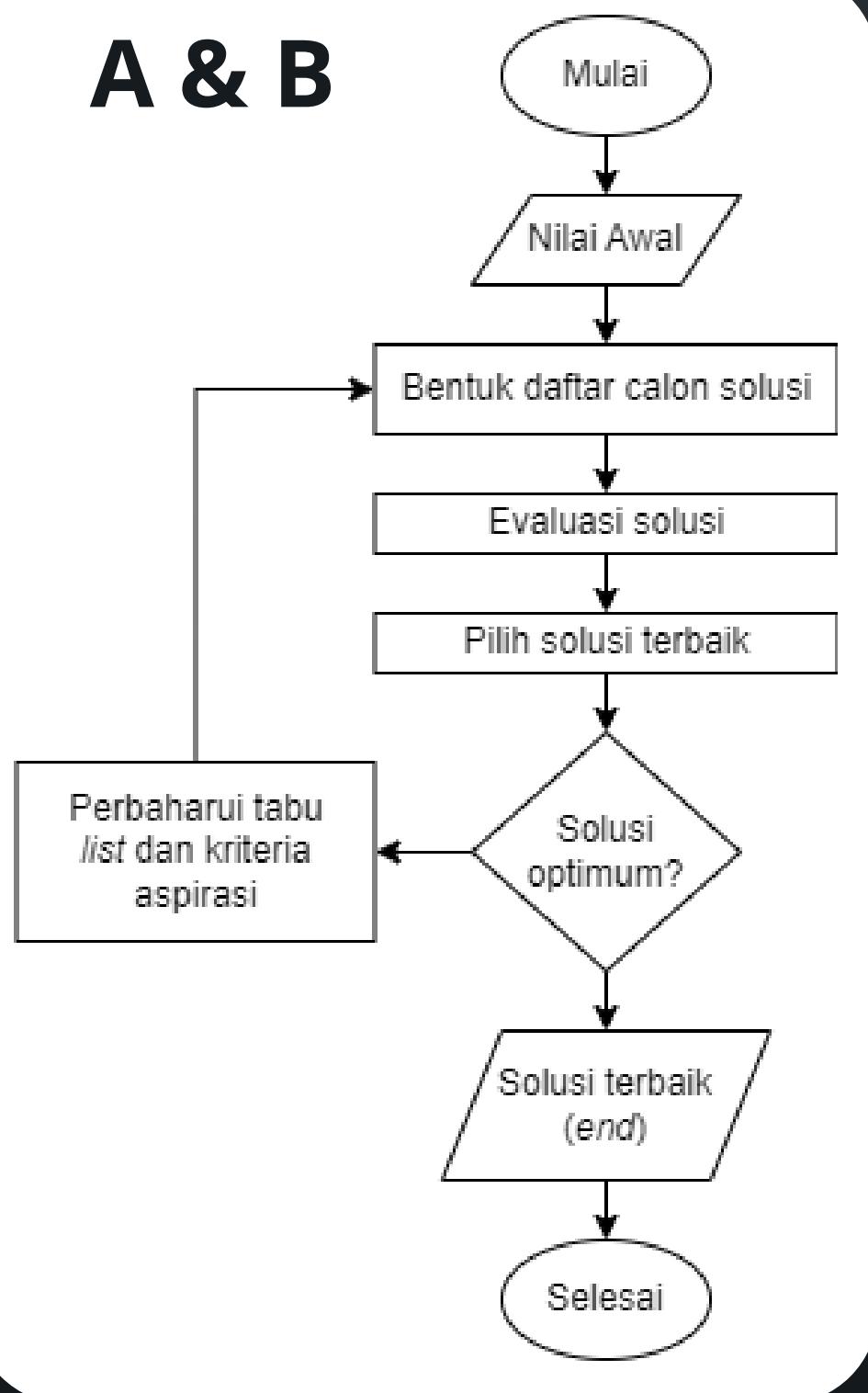
Tabu Search Algorithm: Stopping Conditions

1 $N(i, K + 1) = 0$. (tidak ada solusi layak di lingkungan solusi i)

2 K lebih besar dari jumlah iterasi maksimum yang diperbolehkan.

3 Jumlah iterasi sejak perbaikan terakhir i * lebih besar dari jumlah yang ditentukan.

4 Bukti dapat diberikan bahwa solusi optimal telah diperoleh.

**A & B**

EXPLANATION

1. Menginisialisasi solusi awal (S) dan tabu list (T)
2. Membuat sekumpulan solusi (neighborhood) dari solusi awal (S)
3. Memilih solusi terbaik (S') dari neighborhood sebelumnya
4. Melakukan pengecekan apakah solusi terbaik yang didapatkan sebelumnya berada di tabu list atau tidak [Jika berada di tabu list maka akan dilanjutkan untuk pengecekan aspiration criteria (Jika tidak memenuhi aspiration criteria maka akan delete solusi tersebut dari candidate list), jika tidak maka update solusi awal sebelumnya menjadi solusi terbaik yang baru didapat]
5. Mengecek kriteria penghentian atau stopping conditions



TIME COMPLEXITY

Faktor-faktor utama dalam Time Complexity pada Tabu Search yaitu sebagai berikut.

- Number of Iterations (T)
- Neighborhood Size (N)
- Solution Evaluation Time (E)
- Tabu List Management (L)

Keseluruhan Time Complexity dari Tabu Search:

$$O(T \times (N \times E + L))$$



```

● ○ ●
1 # Fungsi untuk menghitung nilai (fitness) dari sebuah solusi
2 def nilai(solusi):
3     return sum(solusi)
4
5 # Fungsi untuk menghasilkan tetangga dari solusi saat ini
6 def generate_tetangga(solusi_saat_ini):
7     tetangga = []
8     for i in range(len(solusi_saat_ini)):
9         for j in range(i + 1, len(solusi_saat_ini)):
10            tetangga_baru = solusi_saat_ini[:]
11            tetangga_baru[i], tetangga_baru[j] = tetangga_baru[j], tetangga_baru[i]
12            tetangga.append(tetangga_baru)
13    return tetangga
14
15 # Fungsi untuk menjalankan algoritma Tabu Search
16 def TabuSearch(masalah, max_iterasi, tabu_list_size):
17     solusi_saat_ini = masalah[:]
18     solusi_terbaik = solusi_saat_ini[:]
19     nilai_terbaik = nilai(solusi_saat_ini)
20     tabu_list = []
21
22     for i in range(max_iterasi):
23         tetangga = generate_tetangga(solusi_saat_ini)
24         tetangga_terbaik = None
25         nilai_terbaik_tetangga = float('inf')
26
27         for t in tetangga:
28             if nilai(t) < nilai_terbaik_tetangga and t not in tabu_list:
29                 tetangga_terbaik = t
30                 nilai_terbaik_tetangga = nilai(t)
31
32         if tetangga_terbaik is None:
33             break
34
35         solusi_saat_ini = tetangga_terbaik
36         if nilai_terbaik_tetangga < nilai_terbaik:
37             solusi_terbaik = tetangga_terbaik
38             nilai_terbaik = nilai_terbaik_tetangga
39
40         tabu_list.append(solusi_saat_ini)
41         if len(tabu_list) > tabu_list_size:
42             tabu_list.pop(0)
43
44     return solusi_terbaik, nilai_terbaik
45

```

```

46 # Contoh masalah: mencari solusi dengan nilai minimum dalam daftar
47 masalah = [3, 5, 2, 1, 4]
48 max_iterasi = 100
49 tabu_list_size = 10
50
51 solusi_terbaik, nilai_terbaik = TabuSearch(masalah, max_iterasi, tabu_list_size)
52
53 print("Solusi terbaik:", solusi_terbaik)
54 print("Nilai terbaik:", nilai_terbaik)

```

Solusi terbaik: [3, 5, 2, 1, 4]
Nilai terbaik: 15

You are viewing Jackson Lawrence's screen View Options

Google Books

Buku

Hasil: solusi x^* dengan jarak total minimum z^* .

Contoh Tabu Search TSP

Kasus TSP dengan 5 kota, jarak antar kota diketahui sebagai berikut

| d = | 0 | 29.411 | 104.8 | 66.753 | 77.261 |
|--------|--------|--------|--------|--------|--------|
| 29.411 | 0 | 93.493 | 38.013 | 72.546 | 32.865 |
| 104.8 | 93.493 | 0 | 80.895 | 74.726 | 32.865 |
| 66.753 | 38.013 | 80.895 | 0 | 74.726 | 32.865 |
| 77.261 | 73.246 | 32.865 | 74.726 | 0 | 32.865 |

Bangkitkan solusi awal dan hitung jarak total - Misalkan, Rute0=[1 4 3 2 5 1]; Jarak : 391.59

Lakukan pertukaran kota (swap)

Rute1=[1 3 4 2 5 1]; Jarak: 374.16, tukar(4,3)

Rute2=[²5 3 2 4 1]; Jarak: 308.45, tukar (4,5)

Rute3=[1 4 2 3 5 1]; Jarak: 308.45, tukar (2,3)

Rute4=[1 4 3 5 2 1]; Jarak: 283.29, tukar (2,5)

| Tukar | Tambahan jarak |
|-------|----------------|
| 4,3 | -17.43 |
| 4,5 | -83.14 |
| 2,3 | -83.14 |
| 2,5 | -108.30 |

* terbaik.

110

Ignatius Steven

DEVELOPER - LOUIS GABRIEL HERNANDES

A15172 - Oscar - Jhonne

Jackson Lawrence

End

Mute Stop Video Security Participants Chat Share Screen Record Reactions Apps Whiteboards Notes

Links:

- <https://www.geeksforgeeks.org/what-is-tabu-search/>
- https://id.wikipedia.org/wiki/Pencarian_tabu
- <https://www.diklatkerja.com/blog/tabu-search-latar-belakang-penjelasan-tipe-memori-dan-contoh-implementasi>
- https://books.google.co.id/books?id=BELaDwAAQBAJ&printsec=frontcover&hl=id&source=gbs_ge_summary_r&cad=0#v=onepage&q&f=true
- <https://chatgpt.com/share/9be11690-e0c1-4a82-b259-5b0404445751>
- <https://chatgpt.com/share/06b474e8-11b7-41b7-bc0b-9a96cf7f120a>

Paper:

- https://research-dashboard.binus.ac.id/uploads/paper/document/publication/Journal/Inasea/Vol.%202015%20No.%201%20April%202014/02_Hotna%20sitorus.pdf
- <https://lib.unnes.ac.id/7119/1/10485.pdf>
- <https://download.garuda.kemdikbud.go.id/article.php?article=1675130&val=13784&title=PENERAPAN%20ALGORITMA%20TABU%20SEARCH%20UNTUK%20OPENJADWALAN%20MATA%20PELAJARAN%20DI%20SMK%20SWASTA%20PELITA-2%20AEKKANOPAN>
- https://elib.unikom.ac.id/files/disk1/307/jbptunikompp-gdl-restinovri-15327-3-13_babii.pdf

Video:

- <https://youtu.be/A7cTp1Fhg9o?si=6bPkRH0jsJF2d-Zh>
- <https://youtu.be/tIDhFPhrCbU?si=cZTqKlcIPd75jwPt>
- <https://youtu.be/am82jb56slc?si=438aa50cbPModiwq>
- https://youtu.be/saNk8h2KuVE?si=DX_Bn_37aMsG7i8o
- https://youtu.be/bI0SL_ZZNTe?si=c_G2XrE1ZHgJdDRN